



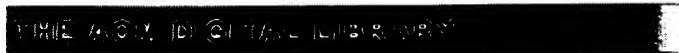
US Patent & Trademark Office

[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

Search: ☒ The ACM Digital Library ☐ The Guide

+automatic <p> +creat??? <p> +form? <p> +validat???

SEARCH



[Feedback](#) [Report a problem](#) [Satisfaction survey](#)

Terms used automatic p creat??? p form? p validat???

Found 16,846 of 147,060

Sort results by

relevance



[Save results to a Binder](#)

[Try an Advanced Search](#)

[Try this search in The ACM Guide](#)

Display results

expanded form



[Search Tips](#)

☐ Open results in a new window

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

Relevance scale ☐ ☐ ☐ ☐ ☐

1 [System Presentation -- CARIBOO: An induction based proof tool for termination with strategies](#)

Olivier Fissore, Isabelle Gnaedig, Hélène Kirchner

October 2002 **Proceedings of the 4th ACM SIGPLAN international conference on Principles and practice of declarative programming**

Full text available: [pdf\(264.09 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

We describe Cariboo, the implementation of an inductive process recently proposed to prove termination of rewriting under strategies on ground term algebras. The method is based on an abstraction mechanism, introducing variables that represent ground terms in normal form, and on narrowing, schematizing reductions on ground terms. It applies in particular to non-terminating systems which are terminating with innermost or local strategies. The narrowing process, well known to easily diverge, is co ...

Keywords: ELAN, constraint, innermost, local strategy, narrowing, ordering, rewriting, rule based language induction, termination

2 [Filter-based model checking of partial systems](#)

Matthew B. Dwyer, Corina S. Pasareanu

November 1998 **ACM SIGSOFT Software Engineering Notes , Proceedings of the 6th ACM SIGSOFT international symposium on Foundations of software engineering**, Volume 23 Issue 6

Full text available: [pdf\(1.43 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Recent years have seen dramatic growth in the application of model checking techniques to the validation and verification of correctness properties of hardware, and more recently software, systems. Most of this work has been aimed at reasoning about properties of complete systems. This paper describes an automatable approach for building finite-state models of partially defined software systems that are amenable to model checking using existing tools. It enables the application of existing model ...

Keywords: assume-guarantee reasoning, filter-based analysis, model checking, software verification and validation

3 [Technical papers: software testing: Automated test case generation for spreadsheets](#)

Marc Fisher, Mingming Cao, Gregg Rothermel, Curtis R. Cook, Margaret M. Burnett

May 2002 **Proceedings of the 24th International Conference on Software Engineering**

Full text available:  [pdf\(1.47 MB\)](#)


Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Spreadsheet languages, which include commercial spreadsheets and various research systems, have had a substantial impact on end-user computing. Research shows, however, that spreadsheets often contain faults. Thus, in previous work, we presented a methodology that assists spreadsheet users in testing their spreadsheet formulas. Our empirical studies have shown that this methodology can help end-users test spreadsheets more adequately and efficiently; however, the process of generating test cases ...

4 Interface design issues for advice-giving expert systems

John M. Carroll, Jean McKendree

January 1987 **Communications of the ACM**, Volume 30 Issue 1

Full text available:  [pdf\(2.28 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Advice giving could become the first successful domain for intelligent interfaces.

5 System validation via constraint modeling

Richard C. Waters

August 1991 **ACM SIGPLAN Notices**, Volume 26 Issue 8


Full text available:  [pdf\(1.30 MB\)](#)

Additional Information: [full citation](#), [index terms](#)

6 Validation of Scientific Programs

William E. Howden

June 1982 **ACM Computing Surveys (CSUR)**, Volume 14 Issue 2

Full text available:  [pdf\(2.92 MB\)](#)

Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

7 Runtime safety analysis of multithreaded programs

Koushik Sen, Grigore Rosu, Gul Agha

September 2003 **ACM SIGSOFT Software Engineering Notes , Proceedings of the 9th European software engineering conference held jointly with 11th ACM SIGSOFT international symposium on Foundations of software engineering**, Volume 28 Issue 5

Full text available:  [pdf\(583.43 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)


Foundational and scalable techniques for runtime safety analysis of multithreaded programs are explored in this paper. A technique based on vector clocks to extract the causal dependency order on state updates from a running multithreaded program is presented, together with algorithms to analyze a multithreaded computation against safety properties expressed using temporal logics. A prototype tool implementing our techniques, is also presented, together with examples where it can predict safety ...

Keywords: JMPaX, Java, LTL, multithreaded program, predictive analysis, runtime monitoring, safety analysis, vector clock

8 Decentralizing a global naming service for improved performance and fault tolerance

D. R. Cheriton, T. P. Mann

May 1989 **ACM Transactions on Computer Systems (TOCS)**, Volume 7 Issue 2

Full text available:  [pdf\(3.19 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

Naming is an important aspect of distributed system design. A naming system allows users and programs to assign character-string names to objects, and subsequently use the names to refer to those objects. With the interconnection of clusters of computers by wide-area

networks and internetworks, the domain over which naming systems must function is growing to encompass the entire world. In this paper we address the problem of a global naming system, proposing a three-level naming ...

9 Structured model specification with a supportive simulation architecture

Edward R. Comer

March 1982 **Proceedings of the 15th annual symposium on Simulation**

Full text available:  pdf(818.35 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Recent emphasis in structured software development has resulted in a greater awareness of an orderly methodology for problem solving. This disciplined approach is adapted for model specification. Through the use of structured software techniques and a Model Specification Language (MSL) a complex distributed computer system model is derived and documented. The model specification is supported by a discrete event simulation architecture which directly reflects the model structure.

10 Declaring and checking non-null types in an object-oriented language

Manuel Fähndrich, K. Rustan M. Leino

October 2003 **ACM SIGPLAN Notices , Proceedings of the 18th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications**, Volume 38 Issue 11

Full text available:  pdf(215.55 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)


Distinguishing non-null references from possibly-null references at the type level can detect null-related errors in object-oriented programs at compile-time. This paper gives a proposal for retrofitting a language such as C# or Java with non-null types. It addresses the central complications that arise in constructors, where declared non-null fields may not yet have been initialized, but the partially constructed object is already accessible. The paper reports experience with an implementation ...

Keywords: Java, c#, non-null types, null references, type system

11 Automatic transformation of series expressions into loops

Richard C. Waters

January 1991 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 13 Issue 1

Full text available:  pdf(3.36 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)


The benefits of programming in a functional style are well known. In particular, algorithms that are expressed as compositions of functions operating on sequences/vectors/streams of data elements are easier to understand and modify than equivalent algorithms expressed as loops. Unfortunately, this kind of expression is not used anywhere near as often as it could be, for at least three reasons: (1) most programmers are less familiar with this kind of expression than with loops; (2) most pro ...

Keywords: sequences, series, streams, vectors

12 Automatic translation of FORTRAN programs to vector form

Randy Allen, Ken Kennedy

October 1987 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 9 Issue 4

Full text available:  pdf(3.14 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)


The recent success of vector computers such as the Cray-1 and array processors such as those manufactured by Floating Point Systems has increased interest in making vector

operations available to the FORTRAN programmer. The FORTRAN standards committee is currently considering a successor to FORTRAN 77, usually called FORTRAN 8x, that will permit the programmer to explicitly specify vector and array operations. Although FORTRAN 8x will make it convenient to specify explicit vector ...

13 Parcel: project for the automatic restructuring and concurrent evaluation of LISP

L. Harrison

June 1988 **Proceedings of the 2nd international conference on Supercomputing**

Full text available:  pdf(1.52 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Parcel (Project for the Automatic Restructuring and Concurrent Evaluation of Lisp) is an investigation of the problem of compiling Lisp for evaluation on a shared memory multiprocessor. In this paper, we present an overview of the process of compilation in Parcel. This process consists, broadly, of an interprocedural analysis, followed by a function-level restructuring of the lambda expressions that constitute a program. We discuss both of these phases, and illustrate the steps of restructu ...

14 A framework for building non-functional software architectures

Nelson S. Rosa, George R. R. Justo, Paulo R. F. Cunha

March 2001 **Proceedings of the 2001 ACM symposium on Applied computing**

Full text available:  pdf(140.64 KB)

Additional Information: [full citation](#), [references](#), [index terms](#)

15 An approach to support automatic generation of user interfaces

Prasun Dewan, Marvin Solomon

October 1990 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 12 Issue 4

Full text available:  pdf(3.55 MB)


Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

In traditional interactive programming environments, each application individually manages its interaction with the human user. The result is duplication of effort in implementing user interface code and nonuniform—hence confusing—input conventions. This paper presents an approach to support automatic generation of user interfaces in environments based on algebraic languages. The approach supports the editing model of interaction, which allows a user to view all appli ...

16 Automatic generation of production rules for integrity maintenance

Stefano Ceri, Piero Fraternali, Stefano Paraboschi, Letizia Tanca

September 1994 **ACM Transactions on Database Systems (TODS)**, Volume 19 Issue 3

Full text available:  pdf(3.42 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

In this article we present an approach to integrity maintenance, consisting of automatically generating production rules for integrity enforcement. Constraints are expressed as particular formulas of Domain Relational Calculus; they are automatically translated into a set of repair actions, encoded as production rules of an active database system. Production rules may be redundant (they enforce the same constraint in different ways) and conflicting (because repairing one constraint may caus ...

Keywords: automatic generation of production rules

17 A scalable formal method for design and automatic checking of user interfaces

Jean Berstel, Stefano Crespi Reghizzi, Gilles Roussel, Pierluigi San Pietro

July 2001 **Proceedings of the 23rd International Conference on Software Engineering**

Full text available:  pdf(237.48 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index](#)



[Publisher Site](#)

[terms](#)

The paper addresses the formal specification, design and implementation of the behavioral component of graphical user interfaces. Dialogs are specified by means of modular, communicating grammars called VEG (Visual Event Grammars), which extend traditional BNF grammars to make the modeling of dialogs more convenient.

A VEG specification is independent of the actual layout of the GUI, but it can be easily integrated with various layout design toolkits. The specification may b ...

Keywords: GUI design, applications of model checking, formal methods, human-computer interaction

18 C-TODOS: an automatic tool for office system conceptual design

B. Pernici, F. Barbic, R. Maiocchi, M. G. Fugini, J. R. Rames, C. Rolland

October 1989 **ACM Transactions on Information Systems (TOIS)**, Volume 7 Issue 4

Full text available: [pdf\(2.88 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

Designers of office information systems, which share various features with information systems and software development, need to carefully consider special issues such as document and communication flows, user roles, user interfaces, and available technology. The ESPRIT Project, Automatic TOOLS for Designing Office Information Systems (TODOS), proposes an integrated environment for office design with tools for requirements collection and analysis, conceptual design, rapid prototy ...

19 Automatic pool allocation for disjoint data structures

Chris Lattner, Vikram Adve

June 2002 **ACM SIGPLAN Notices , Proceedings of the workshop on Memory system performance**, Volume 38 Issue 2 supplement

Full text available: [pdf\(1.48 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

This paper presents an analysis technique and a novel program transformation that can enable powerful optimizations for entire linked data structures. The fully automatic transformation converts ordinary programs to use pool (aka region) allocation for heap-based data structures. The transformation relies on an efficient link-time interprocedural analysis to identify disjoint data structures in the program, to check whether these data structures are accessed in a type-safe manner, and to constru ...

20 Parallel execution of prolog programs: a survey

Gopal Gupta, Enrico Pontelli, Khayri A.M. Ali, Mats Carlsson, Manuel V. Hermenegildo

July 2001 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 23 Issue 4

Full text available: [pdf\(1.95 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Since the early days of logic programming, researchers in the field realized the potential for exploitation of parallelism present in the execution of logic programs. Their high-level nature, the presence of nondeterminism, and their referential transparency, among other characteristics, make logic programs interesting candidates for obtaining speedups through parallel execution. At the same time, the fact that the typical applications of logic programming frequently involve irregular computatio ...

Keywords: Automatic parallelization, constraint programming, logic programming, parallelism, prolog

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2004 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)

[IEEE HOME](#) | [SEARCH IEEE](#) | [SHOP](#) | [WEB ACCOUNT](#) | [CONTACT IEEE](#)

» Se

[Membership](#) [Publications/Services](#) [Standards](#) [Conferences](#) [Careers/Jobs](#)**IEEE Xplore®**
RELEASE 1.8Welcome
United States Patent and Trademark Office[Help
Review](#)[FAQ](#)[Terms](#)[IEEE Peer](#)[Quick Links](#)**Welcome to IEEE Xplore®**

- ☐ Home
- ☐ What Can I Access?
- ☐ Log-out

Tables of Contents

- ☐ Journals & Magazines
- ☐ Conference Proceedings
- ☐ Standards

Search

- ☐ By Author
- ☐ Basic
- ☐ Advanced
- ☐ CrossRef

Member Services

- ☐ Join IEEE
- ☐ Establish IEEE Web Account
- ☐ Access the IEEE Member Digital Library

IEEE Enterprise

- ☐ Access the IEEE Enterprise File Cabinet

[Print Format](#)Your search matched **0** of **1099723** documents.A maximum of **500** results are displayed, **15** to a page, sorted by **Relevance Descending** order.**Refine This Search:**

You may refine your search by editing the current search expression or entering a new one in the text box.

[Search](#)☐ Check to search within this result set**Results Key:****JNL** = Journal or Magazine **CNF** = Conference **STD** = Standard**Results:****No documents matched your query.**[Home](#) | [Log-out](#) | [Journals](#) | [Conference Proceedings](#) | [Standards](#) | [Search by Author](#) | [Basic Search](#) | [Advanced Search](#) | [Join IEEE](#) | [Web Account](#) | [New this week](#) | [OPAC Linking Information](#) | [Your Feedback](#) | [Technical Support](#) | [Email Alerting](#) | [No Robots Please](#) | [Release Notes](#) | [IEEE Online Publications](#) | [Help](#) | [FAQ](#) | [Terms](#) | [Back to Top](#)

Copyright © 2004 IEEE — All rights reserved

[IEEE HOME](#) | [SEARCH IEEE](#) | [SHOP](#) | [WEB ACCOUNT](#) | [CONTACT IEEE](#)

» Se

[Membership](#) | [Publications/Services](#) | [Standards](#) | [Conferences](#) | [Careers/Jobs](#)**IEEE Xplore®**
RELEASE 1.8Welcome
United States Patent and Trademark Office[Help](#)
[Review](#)[FAQ](#)[Terms](#)[IEEE Peer](#)[Quick Links](#)**Welcome to IEEE Xplore®**

- ☐ Home
- ☐ What Can I Access?
- ☐ Log-out

Tables of Contents

- ☐ Journals & Magazines
- ☐ Conference Proceedings
- ☐ Standards

Search

- ☐ By Author
- ☐ Basic
- ☐ Advanced
- ☐ CrossRef

Member Services

- ☐ Join IEEE
- ☐ Establish IEEE Web Account
- ☐ Access the IEEE Member Digital Library

IEEE Enterprise

- ☐ Access the IEEE Enterprise File Cabinet

[Print Format](#)Your search matched **0** of **1099723** documents.A maximum of **500** results are displayed, **15** to a page, sorted by **Relevance Descending** order.**Refine This Search:**

You may refine your search by editing the current search expression or enter a new one in the text box.

☐ Check to search within this result set**Results Key:****JNL** = Journal or Magazine **CNF** = Conference **STD** = Standard**Results:****No documents matched your query.**[Home](#) | [Log-out](#) | [Journals](#) | [Conference Proceedings](#) | [Standards](#) | [Search by Author](#) | [Basic Search](#) | [Advanced Search](#) | [Join IEEE](#) | [Web Account](#) | [New this week](#) | [OPAC Linking Information](#) | [Your Feedback](#) | [Technical Support](#) | [Email Alerting](#) | [No Robots Please](#) | [Release Notes](#) | [IEEE Online Publications](#) | [Help](#) | [FAQ](#) | [Terms](#) | [Back to Top](#)

Copyright © 2004 IEEE — All rights reserved

[IEEE HOME](#) | [SEARCH IEEE](#) | [SHOP](#) | [WEB ACCOUNT](#) | [CONTACT IEEE](#)

» Se

[Membership](#) | [Publications/Services](#) | [Standards](#) | [Conferences](#) | [Careers/Jobs](#)**IEEE Xplore®**
RELEASE 1.8Welcome
United States Patent and Trademark Office[Help](#)
[Review](#)[FAQ](#)[Terms](#)[IEEE Peer](#)[Quick Links](#)**Welcome to IEEE Xplore®**

- ☐ Home
- ☐ What Can I Access?
- ☐ Log-out

Tables of Contents

- ☐ Journals & Magazines
- ☐ Conference Proceedings
- ☐ Standards

Search

- ☐ By Author
- ☐ Basic
- ☐ Advanced
- ☐ CrossRef

Member Services

- ☐ Join IEEE
- ☐ Establish IEEE Web Account
- ☐ Access the IEEE Member Digital Library

IEEE Enterprise

- ☐ Access the IEEE Enterprise File Cabinet

[Print Format](#)Your search matched **0** of **1099723** documents.A maximum of **500** results are displayed, **15** to a page, sorted by **Relevance Descending** order.**Refine This Search:**

You may refine your search by editing the current search expression or entering new one in the text box.

☐ Check to search within this result set**Results Key:****JNL** = Journal or Magazine **CNF** = Conference **STD** = Standard**Results:****No documents matched your query.**[Home](#) | [Log-out](#) | [Journals](#) | [Conference Proceedings](#) | [Standards](#) | [Search by Author](#) | [Basic Search](#) | [Advanced Search](#) | [Join IEEE](#) | [Web Account](#) | [New this week](#) | [OPAC Linking Information](#) | [Your Feedback](#) | [Technical Support](#) | [Email Alerting](#) | [No Robots Please](#) | [Release Notes](#) | [IEEE Online Publications](#) | [Help](#) | [FAQ](#) | [Terms](#) | [Back to Top](#)

Copyright © 2004 IEEE — All rights reserved